Maximal δ-packings for finite VC-dimension set systems and application to discrepancy

Joint work with Victor-Emmanuel Brunel and Nabil Mustafa

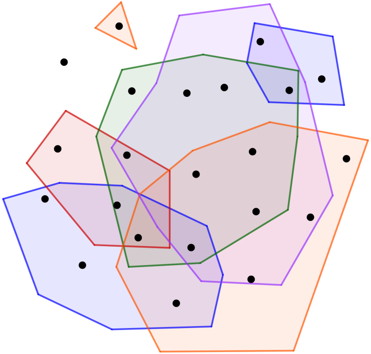
Alexandre Louvet

Université Sorbonne Paris Nord

October 10, 2023

Table of Contents

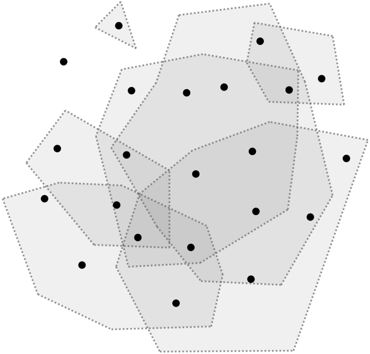
δ-packings



Given $(X, \mathcal{F})$ a set system, a δ-packing is a collection $\mathcal{P} \subseteq \mathcal{F}$ such that:

$$\forall P_1, P_2 \in \mathcal{P}, |\Delta(P_1, P_2)| \geq \delta$$

δ-packings

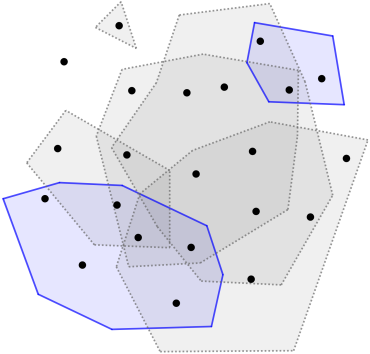


$\delta = 9$

Given $(X, \mathcal{F})$ a set system, a δ-packing is a collection $\mathcal{P} \subseteq \mathcal{F}$ such that:

$$\forall P_1, P_2 \in \mathcal{P}, |\Delta(P_1, P_2)| \geq \delta$$

δ-packings

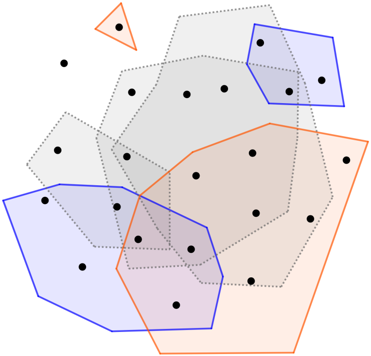


$\delta = 9$

Given $(X, \mathcal{F})$ a set system, a δ-packing is a collection $\mathcal{P} \subseteq \mathcal{F}$ such that:

$$\forall P_1, P_2 \in \mathcal{P}, |\Delta(P_1, P_2)| \geq \delta$$

δ-packings

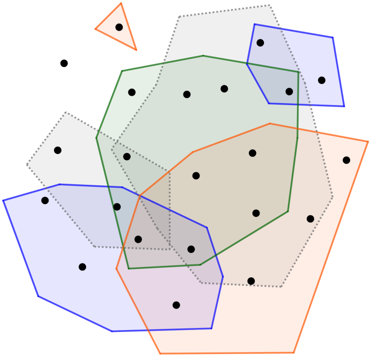


$\delta = 9$

Given $(X, \mathcal{F})$ a set system, a δ-packing is a collection $\mathcal{P} \subseteq \mathcal{F}$ such that:

$$\forall P_1, P_2 \in \mathcal{P}, |\Delta(P_1, P_2)| \geq \delta$$

δ-packings

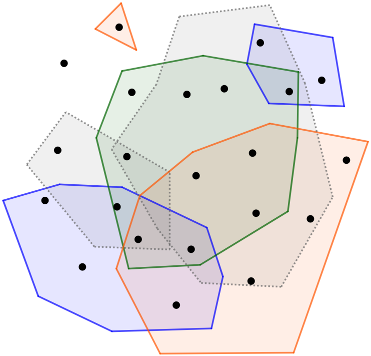


$\delta = 9$

Given $(X, \mathcal{F})$ a set system, a δ-packing is a collection $\mathcal{P} \subseteq \mathcal{F}$ such that:

$$\forall P_1, P_2 \in \mathcal{P}, |\Delta(P_1, P_2)| \geq \delta$$

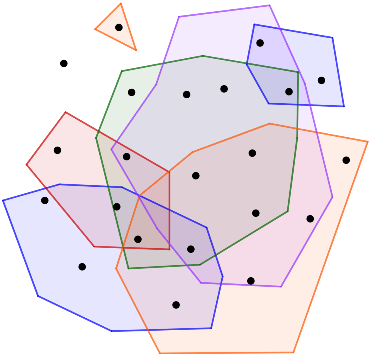
δ-packings



$\delta = 9$

We will study **maximal** packings

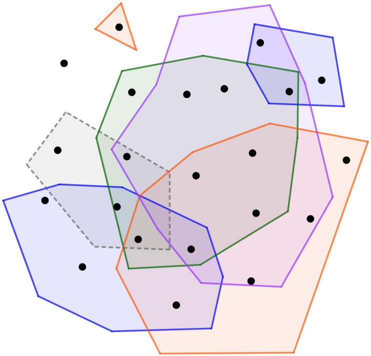
δ-coverings



A δ-covering is a collection $\mathcal{C} \subseteq \mathcal{F}$ such that:

$$\forall F \in \mathcal{F}, \exists C \in \mathcal{C} \text{ s.t. } |\Delta(F, C)| \leq \delta$$

δ-coverings

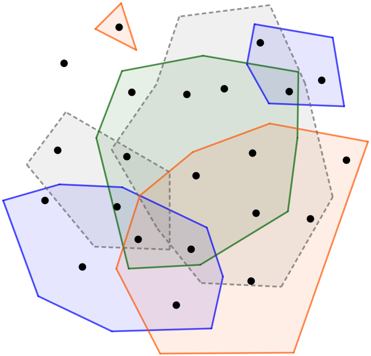


$\delta = 9$

A δ-covering is a collection $\mathcal{C} \subseteq \mathcal{F}$ such that:

$$\forall F \in \mathcal{F}, \exists C \in \mathcal{C} \text{ s.t. } |\Delta(F, C)| \leq \delta$$

δ-coverings

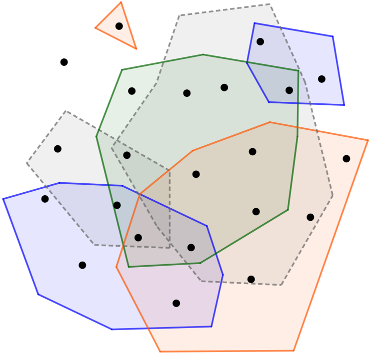


$\delta = 9$

A δ-covering is a collection $\mathcal{C} \subseteq \mathcal{F}$ such that:

$$\forall F \in \mathcal{F}, \exists C \in \mathcal{C} \text{ s.t. } |\Delta(F, C)| \leq \delta$$

δ-coverings



$\delta = 9$

A maximal δ-packing is a minimal δ-covering, however the opposite is not true

How big can it get?

We study the size of a packing for general set systems:

Consider the set system $([n], 2^{[n]})$ and $\delta \in [0, n]$,

How big can it get?

We study the size of a packing for general set systems:

Consider the set system $([n], 2^{[n]})$ and $\delta \in [0, n]$,

- Partition X in $X_1, X_2, \ldots, X_{\frac{n}{\delta}}$ each of size δ

How big can it get?

We study the size of a packing for general set systems:

Consider the set system $([n], 2^{[n]})$ and $\delta \in [0, n]$,

- Partition X in $X_1, X_2, \ldots, X_{\frac{n}{\delta}}$ each of size δ
- Consider the sets $\left\{ \bigcup_{i \in I} X_i : I \subseteq [\frac{n}{\delta}] \right\}$

How big can it get?

We study the size of a packing for general set systems:

Consider the set system $([n], 2^{[n]})$ and $\delta \in [0, n]$,

- Partition X in $X_1, X_2, \ldots, X_{\frac{n}{\delta}}$ each of size δ
- Consider the sets $\left\{ \bigcup_{i \in I} X_i : I \subseteq [\frac{n}{\delta}] \right\}$
- Any two distinct unions are at symmetric difference at least δ

How big can it get?

We study the size of a packing for general set systems:

Consider the set system $([n], 2^{[n]})$ and $\delta \in [0, n]$,

- Partition X in $X_1, X_2, \ldots, X_{\frac{n}{\delta}}$ each of size δ
- Consider the sets $\left\{ \bigcup_{i \in I} X_i : I \subseteq [\frac{n}{\delta}] \right\}$
- Any two distinct unions are at symmetric difference at least δ
- They form a packing of size $2^{\frac{n}{\delta}}$

Problems

- P1: Improving the bound on maximal packing's size (extensively studied)

Problems

- P1: Improving the bound on maximal packing's size (extensively studied)
  - $\rightarrow$ [Hau95] for finite VC-dimension set systems

Problems

- P1: Improving the bound on maximal packing's size (extensively studied)
  - → [Hau95] for finite VC-dimension set systems
  - → [Fox+15] for semi-algebraic set systems

Problems

- P1: Improving the bound on maximal packing's size (extensively studied)
  - → [Hau95] for finite VC-dimension set systems
  - → [Fox+15] for semi-algebraic set systems
  - → [DEG16; Mus16] for set systems with bounded shallow-cell complexity

Problems

- P1: Improving the bound on maximal packing's size (extensively studied)
  - → [Hau95] for finite VC-dimension set systems
  - → [Fox+15] for semi-algebraic set systems
  - → [DEG16; Mus16] for set systems with bounded shallow-cell complexity
- P2: Efficient algorithms to compute approximate maximal packings/minimal coverings

Problems

- P1: Improving the bound on maximal packing's size (extensively studied)
  - $\rightarrow$ [Hau95] for finite VC-dimension set systems
  - $\rightarrow$ [Fox+15] for semi-algebraic set systems
  - $\rightarrow$ [DEG16; Mus16] for set systems with bounded shallow-cell complexity
- P2: Efficient algorithms to compute approximate maximal packings/minimal coverings
  - $\rightarrow$ [MWW93b] finds an approximate δ-covering in time $O\left(\frac{mn^2}{\delta^2}\right)$

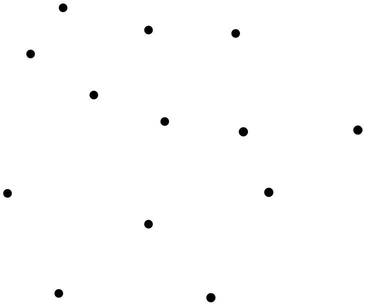
Problems

- P1: Improving the bound on maximal packing's size (extensively studied)
  - $\rightarrow$ [Hau95] for finite VC-dimension set systems
  - $\rightarrow$ [Fox+15] for semi-algebraic set systems
  - $\rightarrow$ [DEG16; Mus16] for set systems with bounded shallow-cell complexity
- P2: Efficient algorithms to compute approximate maximal packings/minimal coverings
  - $\rightarrow$ [MWW93b] finds an approximate δ-covering in time $O\left(\frac{mn^2}{\delta^2}\right)$
  - $\rightarrow$ A greedy algorithm finds a maximal δ-packing in time $O\left(mn\left(\frac{n}{\delta}\right)^d\right)$

Geometric set systems

$(X, \mathcal{F})$ *induced* by halfspaces

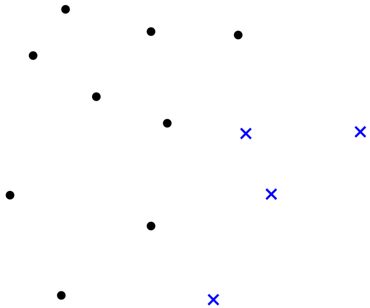
$E \subseteq X$ is in $\mathcal{F}$ iff there exists a halfspace H such that $H \cap X = E$



Geometric set systems

$(X, \mathcal{F})$ *induced* by halfspaces

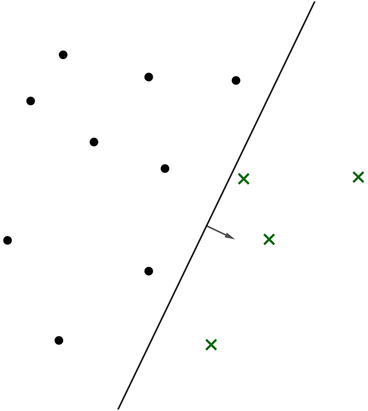
$E \subseteq X$ is in $\mathcal{F}$ iff there exists a halfspace H such that $H \cap X = E$



Geometric set systems

$(X, \mathcal{F})$ *induced* by halfspaces

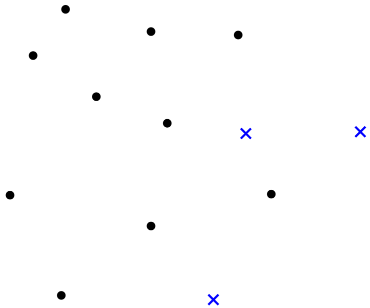
$E \subseteq X$ is in $\mathcal{F}$ iff there exists a halfspace H such that $H \cap X = E$



Geometric set systems

$(X, \mathcal{F})$ *induced* by halfspaces

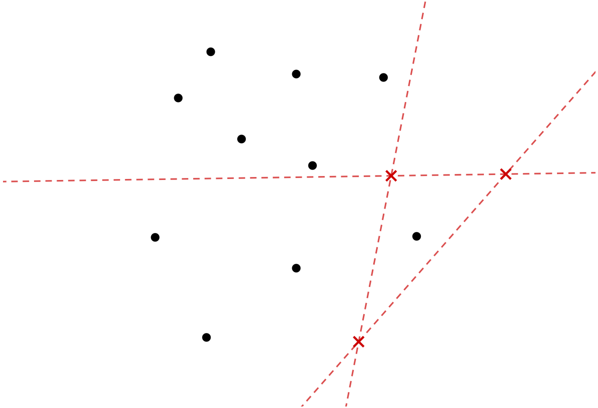
$E \subseteq X$ is in $\mathcal{F}$ iff there exists a halfspace H such that $H \cap X = E$



Geometric set systems

$(X, \mathcal{F})$ *induced* by halfspaces

$E \subseteq X$ is in $\mathcal{F}$ iff there exists a halfspace H such that $H \cap X = E$



Geometric set systems

$(X, \mathcal{F})$ *induced* by halfspaces

$E \subseteq X$ is in $\mathcal{F}$ iff there exists a halfspace H such that $H \cap X = E$

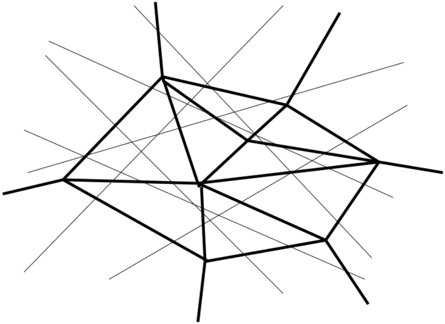
Theorem:

A packing of a set system induced by halfspaces in $\mathbb{R}^d$ has size at most $O\left(\left(\frac{n}{\delta}\right)^d\right)$

Cuttings

We introduce a partition of space called *cuttings* that we will use for the proof of the packing bound.

Given a set H of hyperplanes in E^d, a $\frac{1}{r}$-cutting for H is a collection of d-dimensional simplices with disjoint interiors, together covering E^d and such that the interior of each simplex intersects at most $\frac{|H|}{r}$ hyperplanes.



[Cha01]

Cuttings

We introduce a partition of space called *cuttings* that we will use for the proof of the packing bound.

Given a set H of hyperplanes in E^d, a $\frac{1}{r}$-cutting for H is a collection of d-dimensional simplices with disjoint interiors, together covering E^d and such that the interior of each simplex intersects at most $\frac{|H|}{r}$ hyperplanes.

Theorem [Cha93]

One can construct a $\frac{1}{r}$-cutting of size $O(r^d)$ in time $O(nr^{d-1})$

Proof of the packing bound for halfspaces

Theorem:

A packing of a set system induced by halfspaces in $\mathbb{R}^d$ has size at most $O\left(\left(\frac{n}{\delta}\right)^d\right)$

Proof of the packing bound for halfspaces

Theorem:

A packing of a set system induced by halfspaces in $\mathbb{R}^d$ has size at most $O\left(\left(\frac{n}{\delta}\right)^d\right)$

Proof sketch:

- Given a set system $(X, \mathcal{F})$ consider its dual $(Y, \mathcal{G})$

Proof of the packing bound for halfspaces

Theorem:

A packing of a set system induced by halfspaces in $\mathbb{R}^d$ has size at most $O\left(\left(\frac{n}{\delta}\right)^d\right)$

Proof sketch:

- Given a set system $(X, \mathcal{F})$ consider its dual $(Y, \mathcal{G})$
- A $\frac{\delta}{n}$-cutting of $(Y, \mathcal{G})$ gives a collection of $O\left(\left(\frac{n}{\delta}\right)^d\right)$ each intersected by at most δ hyperplanes

Proof of the packing bound for halfspaces

Theorem:

A packing of a set system induced by halfspaces in $\mathbb{R}^d$ has size at most $O\left(\left(\frac{n}{\delta}\right)^d\right)$

Proof sketch:

- Given a set system $(X, \mathcal{F})$ consider its dual $(Y, \mathcal{G})$
- A $\frac{\delta}{n}$-cutting of $(Y, \mathcal{G})$ gives a collection of $O\left(\left(\frac{n}{\delta}\right)^d\right)$ each intersected by at most δ hyperplanes
- The number of hyperplanes of $\mathcal{G}$ intersecting the segment between two elements of Y is equal to the symmetric difference between the dual of these two elements in $\mathcal{F}$

Proof of the packing bound for halfspaces

Theorem:

A packing of a set system induced by halfspaces in $\mathbb{R}^d$ has size at most $O\left(\left(\frac{n}{\delta}\right)^d\right)$

Proof sketch:

- Given a set system $(X, \mathcal{F})$ consider its dual $(Y, \mathcal{G})$
- A $\frac{\delta}{n}$-cutting of $(Y, \mathcal{G})$ gives a collection of $O\left(\left(\frac{n}{\delta}\right)^d\right)$ each intersected by at most δ hyperplanes
- The number of hyperplanes of $\mathcal{G}$ intersecting the segment between two elements of Y is equal to the symmetric difference between the dual of these two elements in $\mathcal{F}$
- Thus a packing can only contain at most one element from each simplex which gives a bound of $O\left(\left(\frac{n}{\delta}\right)^d\right)$ on its size

Generalization with VC-dimension

We now consider the generalization of this result to combinatorial set systems:

Generalization with VC-dimension

We now consider the generalization of this result to combinatorial set systems:

The VC-dimension of a set system $(X, \mathcal{F})$, is the size of the largest $Y \subseteq X$ for which $|F \cap Y : F \in \mathcal{F}| = 2^{|Y|}$

Generalization with VC-dimension

We now consider the generalization of this result to combinatorial set systems:

The VC-dimension of a set system $(X, \mathcal{F})$, is the size of the largest $Y \subseteq X$ for which $|F \cap Y : F \in \mathcal{F}| = 2^{|Y|}$

Halfspaces

Set system spanned by halfspaces of $\mathbb{R}^d$ have VC-dimension $d+1$

Generalization with VC-dimension

We now consider the generalization of this result to combinatorial set systems:

The VC-dimension of a set system $(X, \mathcal{F})$, is the size of the largest $Y \subseteq X$ for which $|F \cap Y : F \in \mathcal{F}| = 2^{|Y|}$

Halfspaces

Set system spanned by halfspaces of $\mathbb{R}^d$ have VC-dimension $d + 1$

Sauer-Shelah lemma [Sau72; She72]

If $(X, \mathcal{F})$ has VC-dimension $\leq d$, then $|\mathcal{F}| = O(n^d)$

Packing lemma

Packing lemma [Hau95]

Let $(X, \mathcal{F})$ be a set system with finite VC-dimension $\leq d$, then δ-packing of $(X, \mathcal{F})$ has size $O\left(\left(\frac{n}{\delta}\right)^d\right)$

Packing lemma

Packing lemma [Hau95]

Let $(X, \mathcal{F})$ be a set system with finite VC-dimension $\leq d$, then δ-packing of $(X, \mathcal{F})$ has size $O\left(\left(\frac{n}{\delta}\right)^d\right)$

Interpretation:

- The symmetric difference can be seen as a metric for sets

Packing lemma

Packing lemma [Hau95]

Let $(X, \mathcal{F})$ be a set system with finite VC-dimension $\leq d$, then δ-packing of $(X, \mathcal{F})$ has size $O\left(\left(\frac{n}{\delta}\right)^d\right)$

Interpretation:

- The symmetric difference can be seen as a metric for sets
- Then the δ-packing question is similar to packing spheres of radius δ in a sphere of radius n

Packing lemma

Packing lemma [Hau95]

Let $(X, \mathcal{F})$ be a set system with finite VC-dimension $\leq d$, then δ-packing of $(X, \mathcal{F})$ has size $O\left(\left(\frac{n}{\delta}\right)^d\right)$

Interpretation:

- The symmetric difference can be seen as a metric for sets
- Then the δ-packing question is similar to packing spheres of radius δ in a sphere of radius n
- The volume of a sphere of radius δ in $\mathbb{R}^d$ is $O(\delta^d)$, thus the packing would have size $O\left(\left(\frac{n}{\delta}\right)^d\right)$

Packing lemma

Packing lemma [Hau95]

Let $(X, \mathcal{F})$ be a set system with finite VC-dimension $\leq d$, then δ-packing of $(X, \mathcal{F})$ has size $O\left(\left(\frac{n}{\delta}\right)^d\right)$

Interpretation:

- The symmetric difference can be seen as a metric for sets
- Then the δ-packing question is similar to packing spheres of radius δ in a sphere of radius n
- The volume of a sphere of radius δ in $\mathbb{R}^d$ is $O(\delta^d)$, thus the packing would have size $O\left(\left(\frac{n}{\delta}\right)^d\right)$
- This theorem shows that packing sets of VC-dimension d is similar to sphere packing in $\mathbb{R}^d$

Table of Contents

Packing/Covering algorithms for finite VC-dimension

We look at efficient algorithms to compute packings/coverings:

Recall:

Packing/Covering algorithms for finite VC-dimension

We look at efficient algorithms to compute packings/coverings:

Recall:

- A greedy algorithm finds a maximal δ-packing in time $O\left(mn\left(\frac{n}{\delta}\right)^d\right)$

Packing/Covering algorithms for finite VC-dimension

We look at efficient algorithms to compute packings/coverings:

Recall:

- A greedy algorithm finds a maximal δ-packing in time $O\left(mn\left(\frac{n}{\delta}\right)^d\right)$
- [MWW93b] finds a δ-covering of size $O\left(\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)^d\right)$ in time $O\left(\frac{mn^2}{\delta^2}\right)$

Packing/Covering algorithms for finite VC-dimension

We look at efficient algorithms to compute packings/coverings:

Recall:

- A greedy algorithm finds a maximal δ-packing in time $O\left(mn\left(\frac{n}{\delta}\right)^d\right)$
- [MWW93b] finds a δ-covering of size $O\left(\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)^d\right)$ in time $O\left(\frac{mn^2}{\delta^2}\right)$

A *near-maximal* δ-packing of a set system $(X, \mathcal{F})$ is collection $\mathcal{P} \subseteq \mathcal{F}$ such that:

- $\mathcal{P}$ is a δ-packing
- $\mathcal{P}$ is a 3δ-covering

Packing/Covering algorithms for finite VC-dimension

We look at efficient algorithms to compute packings/coverings:

Recall:

- A greedy algorithm finds a maximal δ-packing in time $O\left(mn\left(\frac{n}{\delta}\right)^d\right)$
- [MWW93b] finds a δ-covering of size $O\left(\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)^d\right)$ in time $O\left(\frac{mn^2}{\delta^2}\right)$

A *near-maximal* δ-packing of a set system $(X, \mathcal{F})$ is collection $\mathcal{P} \subseteq \mathcal{F}$ such that:

- $\mathcal{P}$ is a δ-packing
- $\mathcal{P}$ is a 3δ-covering

Theorem

We compute a near-maximal δ-packing in time $\tilde{O}\left(\frac{mn^2}{\delta^2} + \left(\frac{n}{\delta}\right)^{2d+2}\right)$.

Proof: ε − *approximation*

We introduce ε-approximation which we need for our algorithm:

Proof: ε − *approximation*

We introduce ε-approximation which we need for our algorithm:

Given $(X, \mathcal{F})$, an ε-approximation is a set A such that:

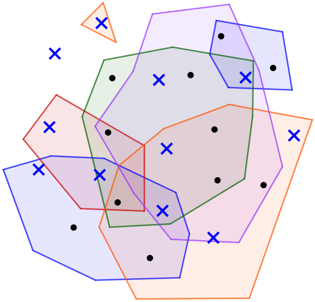
$$\forall F \in \mathcal{F}, \left| \frac{|F|}{|X|} - \frac{|F \cap A|}{|A|} \right| \leq \varepsilon$$

Proof: ε – *approximation*

We introduce ε-approximation which we need for our algorithm:

Given $(X, \mathcal{F})$, an ε-approximation is a set A such that:

$$\forall F \in \mathcal{F}, \left| \frac{|F|}{|X|} - \frac{|F \cap A|}{|A|} \right| \leq \varepsilon$$



$|X| = 22$, $\varepsilon = 1/22$ (no more than one element difference in each set)

Proof: Constructing a packing

We first show how to construct a near-maximal δ-packing:

Proof: Constructing a packing

We first show how to construct a near-maximal δ-packing:

Algorithm Packing($\mathcal{F}$)

$A \leftarrow$ random sample of X of size $O\left(\frac{4n^2}{\delta^2}\right)$
$\mathcal{P} \leftarrow \emptyset$
for $F \in \mathcal{F}$ **do**
  if $\forall P \in \mathcal{P}, |\Delta(P \cap A, F \cap A)| \geq \frac{3n}{2\delta}$ **then**
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{F\}$
  end if
end for
return $\mathcal{P}$

Proof: Constructing a packing

We first show how to construct a near-maximal δ-packing:

Algorithm Packing($\mathcal{F}$)

```
1: A ← random sample of X of size O(4n²/δ²)
2: 𝒫 ← ∅
3: for F ∈ ℱ do
4:     if ∀P ∈ 𝒫, |Δ(P ∩ A, F ∩ A)| ≥ 3n/(2δ) then
5:         𝒫 ← 𝒫 ∪ {F}
6:     end if
7: end for
8: return 𝒫
```

- A is a $\frac{\delta}{2}$-approximation of $(X, \mathcal{F})$

Proof: Constructing a packing

We first show how to construct a near-maximal δ-packing:

Algorithm Packing($\mathcal{F}$)

1: $A \leftarrow$ random sample of X of size $O\left(\frac{4n^2}{\delta^2}\right)$
2: $\mathcal{P} \leftarrow \emptyset$
3: **for** $F \in \mathcal{F}$ **do**
4: **if** $\forall P \in \mathcal{P}, |\Delta(P \cap A, F \cap A)| \geq \frac{3n}{2\delta}$ **then**
5: $\mathcal{P} \leftarrow \mathcal{P} \cup \{F\}$
6: **end if**
7: **end for**
8: **return** $\mathcal{P}$

- A is a $\frac{\delta}{2}$-approximation of $(X, \mathcal{F})$

$\rightarrow$ $\delta = \frac{3\delta}{2} - \frac{\delta}{2} \leq |\Delta(P, F)| \leq \frac{3\delta}{2} + \frac{\delta}{2} = 2\delta$

Proof: Constructing a packing

We first show how to construct a near-maximal δ-packing:

Algorithm Packing($\mathcal{F}$)

$A \leftarrow$ random sample of X of size $O\left(\frac{4n^2}{\delta^2}\right)$
$\mathcal{P} \leftarrow \emptyset$
for $F \in \mathcal{F}$ **do**
  if $\forall P \in \mathcal{P}, |\Delta(P \cap A, F \cap A)| \geq \frac{3n}{2\delta}$ **then**
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{F\}$
  end if
end for
return $\mathcal{P}$

- A is a $\frac{\delta}{2}$-approximation of $(X, \mathcal{F})$

  → $\delta = \frac{3\delta}{2} - \frac{\delta}{2} \leq |\Delta(P, F)| \leq \frac{3\delta}{2} + \frac{\delta}{2} = 2\delta$
  → $\mathcal{P}$ is a δ-packing and a 2δ-covering

Proof: $\varepsilon - nets$

We introduce ε-nets which we need for our algorithm:

Proof: $\varepsilon - nets$

We introduce ε-nets which we need for our algorithm:

Given $(X, \mathcal{F})$, an ε-net is a set N such that:

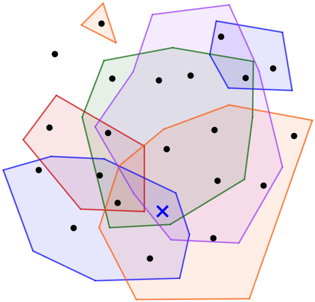
$$\forall F \in \mathcal{F}, |F| \geq \varepsilon|X|, |F \cap N| > 0$$

Proof: $\varepsilon - nets$

We introduce ε-nets which we need for our algorithm:

Given $(X, \mathcal{F})$, an ε-net is a set N such that:

$$\forall F \in \mathcal{F}, |F| \geq \varepsilon|X|, |F \cap N| > 0$$



$|X| = 22$, $\varepsilon = 5/22$ (sets with 5 elements and more)

Proof: An algorithm for small δ-covering [MWW93b]

We show how to construct a small δ-covering:

Proof: An algorithm for small δ-covering [MWW93b]

We show how to construct a small δ-covering:

Algorithm δ-covering algorithm

$\mathcal{S} \leftarrow \emptyset$
$\mathcal{C} \leftarrow \emptyset$
$N \leftarrow$ uniform random sample of X of size $O\left(\frac{n}{\delta} \log\left(\frac{n}{\delta}\right)\right)$
for $F \in \mathcal{F}$ **do**
    $Q \leftarrow F \cap N$
    if $Q \notin \mathcal{S}$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{Q\}$
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{F\}$
    end if
end for
return $(\mathcal{C})$

Proof: An algorithm for small δ-covering [MWW93b]

We show how to construct a small δ-covering:

Algorithm δ-covering algorithm

$\mathcal{S} \leftarrow \emptyset$
$\mathcal{C} \leftarrow \emptyset$
$N \leftarrow$ uniform random sample of X of size $O\left(\frac{n}{\delta} \log\left(\frac{n}{\delta}\right)\right)$
for $F \in \mathcal{F}$ **do**
    $Q \leftarrow F \cap N$
    if $Q \notin \mathcal{S}$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{Q\}$
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{F\}$
    end if
end for
return $(\mathcal{C})$

- N is a $\frac{\delta}{n}$-net, i.e. sets bigger than δ contain an element of N

Proof: An algorithm for small δ-covering [MWW93b]

We show how to construct a small δ-covering:

Algorithm δ-covering algorithm

$\mathcal{S} \leftarrow \emptyset$
$\mathcal{C} \leftarrow \emptyset$
$N \leftarrow$ uniform random sample of X of size $O\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)$
for $F \in \mathcal{F}$ **do**
    $Q \leftarrow F \cap N$
    if $Q \notin \mathcal{S}$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{Q\}$
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{F\}$
    end if
end for
return $(\mathcal{C})$

- N is a $\frac{\delta}{n}$-net, i.e. sets bigger than δ contain an element of N
- If $F, F' \in \mathcal{F}$ are such that $F \cap N = F' \cap N$, then $|\Delta(F, F')| \leq \delta$

Proof: An algorithm for small δ-covering [MWW93b]

We show how to construct a small δ-covering:

Algorithm δ-covering algorithm

```
S ← ∅
C ← ∅
N ← uniform random sample of X of size O((n/δ) log(n/δ))
for F ∈ 𝓕 do
    Q ← F ∩ N
    if Q ∉ S then
        S ← S ∪ {Q}
        C ← C ∪ {F}
    end if
end for
return (C)
```

- N is a $\frac{\delta}{n}$-net, i.e. sets bigger than δ contain an element of N
- If $F, F' \in \mathcal{F}$ are such that $F \cap N = F' \cap N$, then $|\Delta(F, F')| \leq \delta$

By Sauer-Shelah's lemma,
$|\mathcal{C}| = O\left(|N|^d\right) = O\left(\left(\frac{n}{\delta}\right)^d \log^d\left(\frac{n}{\delta}\right)\right)$

Proof: Combining the two ideas

- Compute a $O\left(\left(\frac{n}{\delta}\right)^d \log^d\left(\frac{n}{\delta}\right)\right)$-size δ-covering $\mathcal{C}$. This takes time $O\left(\frac{mn^2}{\delta^2} + \left(\frac{n}{\delta}\right)^d \log^d\left(\frac{n}{\delta}\right)\right)$

Proof: Combining the two ideas

- Compute a $O\left(\left(\frac{n}{\delta}\right)^d \log^d\left(\frac{n}{\delta}\right)\right)$-size δ-covering $\mathcal{C}$. This takes time $O\left(\frac{mn^2}{\delta^2} + \left(\frac{n}{\delta}\right)^d \log^d\left(\frac{n}{\delta}\right)\right)$
- Apply the packing algorithm on $\mathcal{C}$ to obtain a δ-packing $\mathcal{P}$. The result is a 3δ-covering since $\forall F \in \mathcal{F}, \exists C \in \mathcal{C}$ s.t. $\Delta(F, C) \leq \delta$ and $\forall C \in \mathcal{C}, \exists P \in \mathcal{P}$ s.t. $\Delta(C, P) \leq 2\delta$

Proof: Combining the two ideas

- Compute a $O\left(\left(\frac{n}{\delta}\right)^d \log^d\left(\frac{n}{\delta}\right)\right)$-size δ-covering $\mathcal{C}$. This takes time $O\left(\frac{mn^2}{\delta^2} + \left(\frac{n}{\delta}\right)^d \log^d\left(\frac{n}{\delta}\right)\right)$
- Apply the packing algorithm on $\mathcal{C}$ to obtain a δ-packing $\mathcal{P}$. The result is a 3δ-covering since $\forall F \in \mathcal{F}, \exists C \in \mathcal{C}$ s.t. $\Delta(F, C) \leq \delta$ and $\forall C \in \mathcal{C}, \exists P \in \mathcal{P}$ s.t. $\Delta(C, P) \leq 2\delta$
- This takes time $O\left(\left(\frac{n}{\delta}\right)^{2d+2} \log^d\left(\frac{n}{\delta}\right)\right)$ giving total runtime of $\tilde{O}\left(\frac{mn^2}{\delta^2} + \left(\frac{n}{\delta}\right)^{2d+2}\right)$

Table of Contents

Set discrepancy

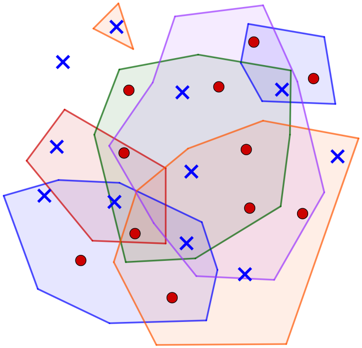
We want to compute a 2-coloring $\chi : X \to \{-1, 1\}$ s.t. $\forall F \in \mathcal{F}, \chi(F) = \sum_{x \in F} \chi(x)$ is small. We call discrepancy and denote:

$$\text{disc}(X, \mathcal{F}) = \max_{F \in \mathcal{F}} |\chi(F)|$$

Set discrepancy

We want to compute a 2-coloring $\chi : X \to \{-1, 1\}$ s.t. $\forall F \in \mathcal{F}, \chi(F) = \sum_{x \in F} \chi(x)$ is small. We call discrepancy and denote:

$$\mathrm{disc}(X, \mathcal{F}) = \max_{F \in \mathcal{F}} |\chi(F)|$$



Previous bounds

- Matousek [Mat95] proved that for VC-dimension $\leq d$ set systems $\mathrm{disc}((X, \mathcal{F})) = O(n^{1/2-1/2d})$. This result is non-constructive and is obtained using chaining.

Previous bounds

- Matousek [Mat95] proved that for VC-dimension $\leq d$ set systems $\text{disc}((X, \mathcal{F})) = O(n^{1/2-1/2d})$. This result is non-constructive and is obtained using chaining.
- Lovett and Meka [LM12] obtain optimal discrepancy for general set systems when $m = n$ in time $\tilde{O}((m+n)^3)$

Previous bounds

- Matousek [Mat95] proved that for VC-dimension $\leq d$ set systems $\text{disc}((X, \mathcal{F})) = O(n^{1/2-1/2d})$. This result is non-constructive and is obtained using chaining.
- Lovett and Meka [LM12] obtain optimal discrepancy for general set systems when $m = n$ in time $\tilde{O}((m+n)^3)$
- Matousek,Welzl and Wernisch [MWW93a]'s cover algorithm used with [LM12] gives discrepancy $O\left(n^{1/2-1/2d}\log^{3/2}(n)\right)$ in time $O\left(mn^{2/d} + n^3\log^{3d}(n)\right)$

Previous bounds

- Matousek [Mat95] proved that for VC-dimension $\leq d$ set systems $\text{disc}((X, \mathcal{F})) = O(n^{1/2-1/2d})$. This result is non-constructive and is obtained using chaining.
- Lovett and Meka [LM12] obtain optimal discrepancy for general set systems when $m = n$ in time $\tilde{O}((m+n)^3)$
- Matousek,Welzl and Wernisch [MWW93a]'s cover algorithm used with [LM12] gives discrepancy $O\left(n^{1/2-1/2d}\log^{3/2}(n)\right)$ in time $O\left(mn^{2/d} + n^3\log^{3d}(n)\right)$
- The greedy packing algorithm with [LM12] gives discrepancy $O\left(n^{1/2-1/2d}\log^{1/2}(n)\right)$ in time $O\left(mn^2 + n^3\right)$

Previous bounds

- Matousek [Mat95] proved that for VC-dimension $\leq d$ set systems $\text{disc}((X, \mathcal{F})) = O(n^{1/2-1/2d})$. This result is non-constructive and is obtained using chaining.
- Lovett and Meka [LM12] obtain optimal discrepancy for general set systems when $m = n$ in time $\tilde{O}((m+n)^3)$
- Matousek,Welzl and Wernisch [MWW93a]'s cover algorithm used with [LM12] gives discrepancy $O\left(n^{1/2-1/2d}\log^{3/2}(n)\right)$ in time $O\left(mn^{2/d} + n^3\log^{3d}(n)\right)$
- The greedy packing algorithm with [LM12] gives discrepancy $O\left(n^{1/2-1/2d}\log^{1/2}(n)\right)$ in time $O\left(mn^2 + n^3\right)$

Theorem:

Given a set system of VC-dimension $\leq d$, we compute a coloring with discrepancy $O\left(n^{1/2-1/2d}\right)$ in time $O\left(mn^{2/d} + n^3\log^{3d}(n)\right)$

Proof: Sub-optimal result

We first show how to obtain discrepancy $O(n^{1/2-1/2d}\sqrt{\log(m)})$

Proof: Sub-optimal result

We first show how to obtain discrepancy $O(n^{1/2-1/2d}\sqrt{\log(m)})$

- Given two sets F, F' : $\mathrm{disc}(F) \leq \mathrm{disc}(F') + \mathrm{disc}(F \backslash F') + \mathrm{disc}(F' \backslash F)$

Proof: Sub-optimal result

We first show how to obtain discrepancy $O(n^{1/2-1/2d}\sqrt{\log(m)})$

- Given two sets F, F' : $\text{disc}(F) \leq \text{disc}(F') + \text{disc}(F \setminus F') + \text{disc}(F' \setminus F)$
- We compute a near-maximal $n^{1-1/d}$-packing and use [LM12]'s algorithm such that all set in the packing have discrepancy $O(n^{1/2-1/2d})$

Proof: Sub-optimal result

We first show how to obtain discrepancy $O(n^{1/2-1/2d}\sqrt{\log(m)})$

- Given two sets F, F' : $\text{disc}(F) \leq \text{disc}(F') + \text{disc}(F \setminus F') + \text{disc}(F' \setminus F)$
- We compute a near-maximal $n^{1-1/d}$-packing and use [LM12]'s algorithm such that all set in the packing have discrepancy $O(n^{1/2-1/2d})$
- [LM12] gives $\sqrt{|S| \log(m)}$ discrepancy on a set S not used as constraint. Thus the symmetric difference sets have discrepancy $\sqrt{n^{1-1/d} \log(m)}$

Proof: Sub-optimal result

We first show how to obtain discrepancy $O(n^{1/2-1/2d}\sqrt{\log(m)})$

- Given two sets F, F' : $\text{disc}(F) \leq \text{disc}(F') + \text{disc}(F \setminus F') + \text{disc}(F' \setminus F)$
- We compute a near-maximal $n^{1-1/d}$-packing and use [LM12]'s algorithm such that all set in the packing have discrepancy $O(n^{1/2-1/2d})$
- [LM12] gives $\sqrt{|S| \log(m)}$ discrepancy on a set S not used as constraint. Thus the symmetric difference sets have discrepancy $\sqrt{n^{1-1/d} \log(m)}$
- This gives $\text{disc}(F) \leq$ $\text{disc}(P) + \text{disc}(P \setminus F) + \text{disc}(F \setminus P) = O\left(\sqrt{n^{1-1/d} \log(m)}\right)$

Proof: Optimality with chaining

We now show how to use chaining to get rid of the extra $\sqrt{\log(m)}$ factor in the discrepancy bound.

Proof: Optimality with chaining

We now show how to use chaining to get rid of the extra $\sqrt{\log(m)}$ factor in the discrepancy bound.

- We compute near-maximal $\frac{n^{1-1/d}}{2^i}$-packing $\mathcal{P}_i$ (for $i \in [0, \log\log m]$)

Proof: Optimality with chaining

We now show how to use chaining to get rid of the extra $\sqrt{\log(m)}$ factor in the discrepancy bound.

- We compute near-maximal $\frac{n^{1-1/d}}{2^i}$-packing $\mathcal{P}_i$ (for $i \in [0, \log \log m]$)
- $\text{disc}(F) \leq \text{disc}(P_0) + \text{disc}(\Delta(P_0, P_1)) + \ldots + \text{disc}(\Delta(P_{\log \log m - 1}, P_{\log \log m})) + \text{disc}(E)$, where $|E| \leq \frac{n^{1-1/d}}{\log(m)}$

Proof: Optimality with chaining

We now show how to use chaining to get rid of the extra $\sqrt{\log(m)}$ factor in the discrepancy bound.

- We compute near-maximal $\frac{n^{1-1/d}}{2^i}$-packing $\mathcal{P}_i$ (for $i \in [0, \log\log m]$)
- $\text{disc}(F) \leq \text{disc}(P_0) + \text{disc}(\Delta(P_0, P_1)) + \ldots + \text{disc}(\Delta(P_{\log\log m - 1}, P_{\log\log m})) + \text{disc}(E)$, where $|E| \leq \frac{n^{1-1/d}}{\log(m)}$
- E has discrepancy $O(n^{1/2-1/2d})$ and by carefully choosing the parameters to use [LM12], so does $\sum_{i=1}^{\log\log(m)} \text{disc}(\Delta(P_{i-1}, P_i))$

Proof: Optimality with chaining

We now show how to use chaining to get rid of the extra $\sqrt{\log(m)}$ factor in the discrepancy bound.

- We compute near-maximal $\frac{n^{1-1/d}}{2^i}$-packing $\mathcal{P}_i$ (for $i \in [0, \log\log m]$)
- $\mathrm{disc}(F) \leq \mathrm{disc}(P_0) + \mathrm{disc}(\Delta(P_0, P_1)) + \ldots + \mathrm{disc}(\Delta(P_{\log\log m - 1}, P_{\log\log m})) + \mathrm{disc}(E)$, where $|E| \leq \frac{n^{1-1/d}}{\log(m)}$
- E has discrepancy $O(n^{1/2-1/2d})$ and by carefully choosing the parameters to use [LM12], so does $\sum_{i=1}^{\log\log(m)} \mathrm{disc}(\Delta(P_{i-1}, P_i))$
- We use [LM12] with n points and $\sum_{i=0}^{\log\log m} |\mathcal{P}_i| = n \log(n)$ sets which gives runtime $O(n^3 \log^{3d}(n))$

Results on ε-approximation

- By computing iterated discrepancy, we obtain an ε-approximation of size $O\left(\varepsilon^{-\frac{2d}{d+1}}\right)$ in time $O\left(\left[mn^{2/d} + n^3 \log^{3d}(n)\right] \log\left(\frac{1}{\varepsilon}\right)\right)$

Results on ε-approximation

- By computing iterated discrepancy, we obtain an ε-approximation of size $O\left(\varepsilon^{-\frac{2d}{d+1}}\right)$ in time $O\left(\left[mn^{2/d} + n^3 \log^{3d}(n)\right] \log\left(\frac{1}{\varepsilon}\right)\right)$
- Using the *merge and reduce* framework [CM96], this gives an ε-approximation of size $O\left(\varepsilon^{-\frac{2d}{d+1}}\right)$ in time $\tilde{O}\left(\frac{n}{\varepsilon^{2d}}\right)$

Open problems

- Is it possible to find a polynomial-time algorithm for (near-)maximal packings with degree $< d$ (for δ bigger than a constant)?

Open problems

- Is it possible to find a polynomial-time algorithm for (near-)maximal packings with degree $< d$ (for δ bigger than a constant)?
- Is it possible to find faster algorithm set systems with smaller packing size than Haussler's bound?

References I

[Sau72] N Sauer. "On the density of families of sets". en. In: *Journal of Combinatorial Theory, Series A* 13.1 (July 1972), pp. 145–147. ISSN: 0097-3165. DOI: 10.1016/0097-3165(72)90019-2.

[She72] Saharon Shelah. "A combinatorial problem; stability and order for models and theories in infinitary languages". In: *Pacific Journal of Mathematics* 41.1 (1972), pp. 247–261.

[Cha93] Bernard Chazelle. "Cutting hyperplanes for divide-and-conquer". en. In: *Discrete & Computational Geometry* 9.2 (Feb. 1993), pp. 145–158. ISSN: 0179-5376, 1432-0444. DOI: 10.1007/BF02189314.

References II

[MWW93a] Jiri Matousek, Emo Welzl, and Lorenz Wernisch. "Discrepancy and approximations for bounded VC-dimension". en. In: *Combinatorica* 13.4 (Dec. 1993), pp. 455–466. ISSN: 1439-6912. DOI: 10.1007/BF01303517.

[MWW93b] Jiří Matoušek, Emo Welzl, and Lorenz Wernisch. "Discrepancy and approximations for bounded VC-dimension". en. In: *Combinatorica* 13.4 (Dec. 1993), pp. 455–466. ISSN: 1439-6912. DOI: 10.1007/BF01303517.

[Hau95] David Haussler. "Sphere packing numbers for subsets of the Boolean n-cube with bounded Vapnik-Chervonenkis dimension". en. In: *Journal of Combinatorial Theory, Series A* 69.2 (Feb. 1995), pp. 217–232. ISSN: 0097-3165. DOI: 10.1016/0097-3165(95)90052-7.

References III

[Mat95] J. Matousek. "Tight upper bounds for the discrepancy of half-spaces". en. In: *Discrete & Computational Geometry* 13.3 (June 1995), pp. 593–601. ISSN: 1432-0444. DOI: 10.1007/BF02574066.

[CM96] Bernard Chazelle and Jiří Matoušek. "On Linear-Time Deterministic Algorithms for Optimization Problems in Fixed Dimension". en. In: *Journal of Algorithms* 21.3 (Nov. 1996), pp. 579–597. ISSN: 01966774. DOI: 10.1006/jagm.1996.0060.

[Cha01] Bernard Chazelle. *The discrepancy method: randomness and complexity*. 2001.

References IV

[LM12] Shachar Lovett and Raghu Meka. *Constructive Discrepancy Minimization by Walking on The Edges*. arXiv:1203.5747 [cs, math] version: 1. Mar. 2012. DOI: 10.48550/arXiv.1203.5747. URL: http://arxiv.org/abs/1203.5747 (visited on 09/01/2022).

[Fox+15] Jacob Fox et al. "A semi-algebraic version of Zarankiewicz's problem". In: arXiv:1407.5705 (Nov. 2015). arXiv:1407.5705 [math]. URL: http://arxiv.org/abs/1407.5705.

[DEG16] Kunal Dutta, Esther Ezra, and Arijit Ghosh. "Two Proofs for Shallow Packings". en. In: *Discrete & Computational Geometry* 56.4 (Dec. 2016), pp. 910–939. ISSN: 1432-0444. DOI: 10.1007/s00454-016-9824-0.

References V

[Mus16] Nabil H. Mustafa. "A Simple Proof of the Shallow Packing Lemma". en. In: *Discrete & Computational Geometry* 55.3 (Apr. 2016), pp. 739–743. ISSN: 1432-0444. DOI: 10.1007/s00454-016-9767-5.

The optimization setup

- Consider the coloring χ as a vector x of $\{-1,1\}^n$ (according to some ordering of X)

The optimization setup

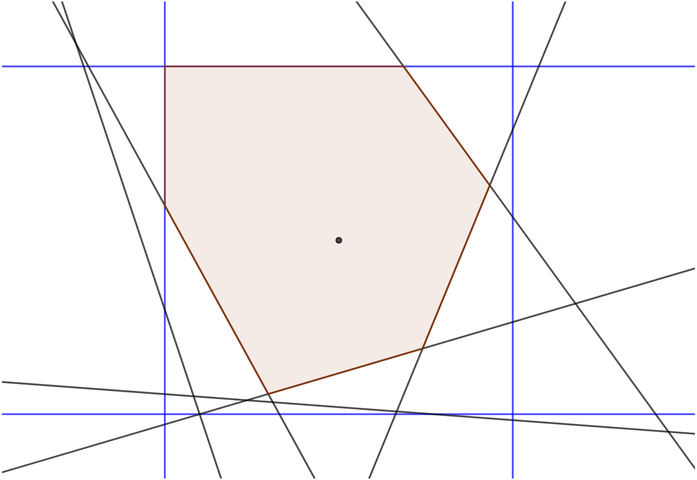
- Consider the coloring χ as a vector x of $\{-1, 1\}^n$ (according to some ordering of X)
- Consider each element of $\mathcal{F}$ as its indicator vector v_F (according to the same ordering)

The optimization setup

- Consider the coloring χ as a vector x of $\{-1, 1\}^n$ (according to some ordering of X)
- Consider each element of $\mathcal{F}$ as its indicator vector v_F (according to the same ordering)
- Then $|\chi(F)| = \langle x, v_F \rangle$

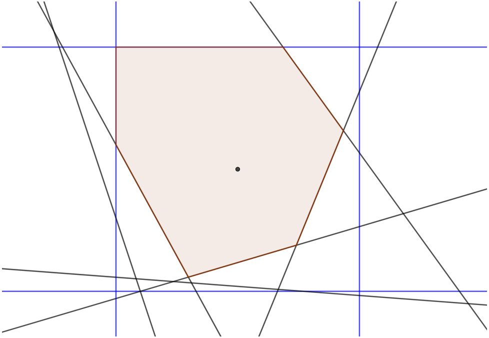
[LM12] random walk and entropy

- The algorithm is a random walk in the polytope formed by the intersection of $\{-1,1\}^n$ and halfspaces defined by $\langle x, v_F \rangle \leq \lambda_F$



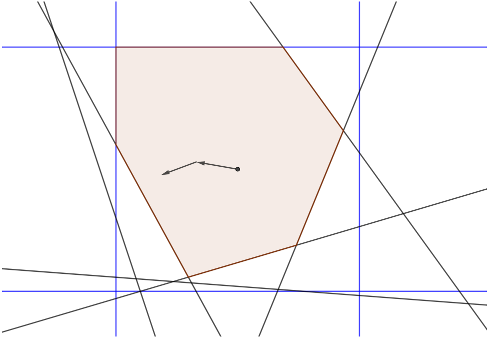
[LM12] random walk and entropy

- The algorithm is a random walk in the polytope formed by the intersection of $\{-1, 1\}^n$ and halfspaces defined by $\langle x, v_F \rangle \leq \lambda_F$



[LM12] random walk and entropy

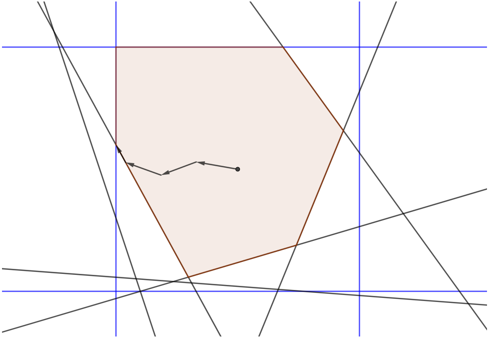
- The algorithm is a random walk in the polytope formed by the intersection of $\{-1,1\}^n$ and halfspaces defined by $\langle x, v_F \rangle \leq \lambda_F$



- The walk simply progresses by moving along unit-size gaussian vectors in each direction

[LM12] random walk and entropy

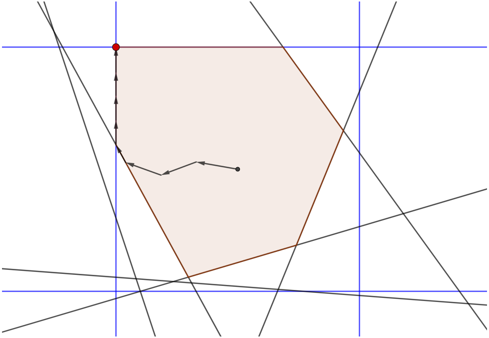
- The algorithm is a random walk in the polytope formed by the intersection of $\{-1,1\}^n$ and halfspaces defined by $\langle x, v_F \rangle \leq \lambda_F$



- The walk simply progresses by moving along unit-size gaussian vectors in each direction
- The walk moves along the constraints hit

[LM12] random walk and entropy

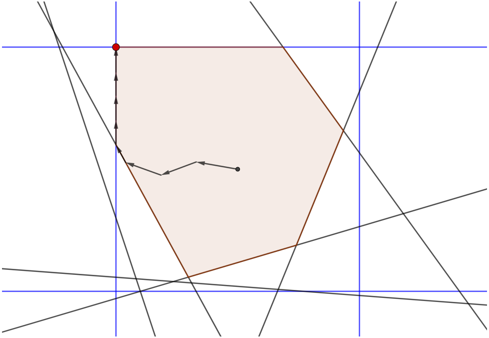
- The algorithm is a random walk in the polytope formed by the intersection of $\{-1, 1\}^n$ and halfspaces defined by $\langle x, v_F \rangle \leq \lambda_F$



- The walk simply progresses by moving along unit-size gaussian vectors in each direction
- The walk moves along the constraints hit
- The goal is to reach a corner of the cube

[LM12] random walk and entropy

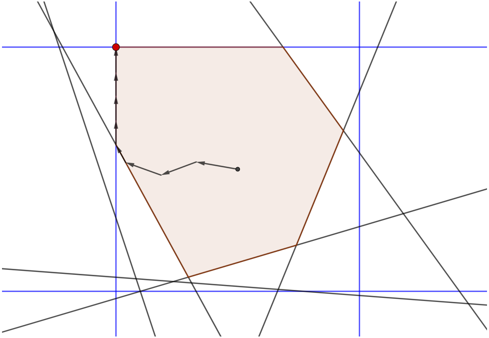
- The algorithm is a random walk in the polytope formed by the intersection of $\{-1, 1\}^n$ and halfspaces defined by $\langle x, v_F \rangle \leq \lambda_F$



- The walk simply progresses by moving along unit-size gaussian vectors in each direction
- The walk moves along the constraints hit
- The goal is to reach a corner of the cube
- If too many constraints are too close to the center it won't be possible

[LM12] random walk and entropy

- The algorithm is a random walk in the polytope formed by the intersection of $\{-1, 1\}^n$ and halfspaces defined by $\langle x, v_F \rangle \leq \lambda_F$



- The walk simply progresses by moving along unit-size gaussian vectors in each direction
- The walk moves along the constraints hit
- The goal is to reach a corner of the cube
- If too many constraints are too close to the center it won't be possible

Entropy condition

$$\sum_{F \in \mathcal{F}} e^{-\lambda_F{}^2} \leq n$$